

```
In [1]: import pandas as pd
import pickle
import warnings
warnings.filterwarnings("ignore")
```

```
In [3]: a=pd.read_csv("C:\\Users\\reshma_koduri\\OneDrive\\Documents\\Advertising.csv")
a
```

```
Out[3]:
```

	Unnamed: 0	TV	radio	newspaper	sales
0	1	230.1	37.8	69.2	22.1
1	2	44.5	39.3	45.1	10.4
2	3	17.2	45.9	69.3	9.3
3	4	151.5	41.3	58.5	18.5
4	5	180.8	10.8	58.4	12.9
...
195	196	38.2	3.7	13.8	7.6
196	197	94.2	4.9	8.1	9.7
197	198	177.0	9.3	6.4	12.8
198	199	283.6	42.0	66.2	25.5
199	200	232.1	8.6	8.7	13.4

200 rows × 5 columns

```
In [4]: a.head(5)
```

```
Out[4]:
```

	Unnamed: 0	TV	radio	newspaper	sales
0	1	230.1	37.8	69.2	22.1
1	2	44.5	39.3	45.1	10.4
2	3	17.2	45.9	69.3	9.3
3	4	151.5	41.3	58.5	18.5
4	5	180.8	10.8	58.4	12.9

```
In [5]: a.tail(5)
```

```
Out[5]:
```

	Unnamed: 0	TV	radio	newspaper	sales
195	196	38.2	3.7	13.8	7.6
196	197	94.2	4.9	8.1	9.7
197	198	177.0	9.3	6.4	12.8
198	199	283.6	42.0	66.2	25.5
199	200	232.1	8.6	8.7	13.4

In [6]:

```
a.describe()
```

Out[6]:

	Unnamed: 0	TV	radio	newspaper	sales
count	200.000000	200.000000	200.000000	200.000000	200.000000
mean	100.500000	147.042500	23.264000	30.554000	14.022500
std	57.879185	85.854236	14.846809	21.778621	5.217457
min	1.000000	0.700000	0.000000	0.300000	1.600000
25%	50.750000	74.375000	9.975000	12.750000	10.375000
50%	100.500000	149.750000	22.900000	25.750000	12.900000
75%	150.250000	218.825000	36.525000	45.100000	17.400000
max	200.000000	296.400000	49.600000	114.000000	27.000000

In [7]:

```
a.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Unnamed: 0      200 non-null    int64
1   TV              200 non-null    float64
2   radio           200 non-null    float64
3   newspaper       200 non-null    float64
4   sales           200 non-null    float64
dtypes: float64(4), int64(1)
memory usage: 7.9 KB
```

In [8]:

```
a['TV'].unique()
```

Out[8]:

```
array([230.1, 44.5, 17.2, 151.5, 180.8, 8.7, 57.5, 120.2, 8.6,
       199.8, 66.1, 214.7, 23.8, 97.5, 204.1, 195.4, 67.8, 281.4,
       69.2, 147.3, 218.4, 237.4, 13.2, 228.3, 62.3, 262.9, 142.9,
       240.1, 248.8, 70.6, 292.9, 112.9, 97.2, 265.6, 95.7, 290.7,
       266.9, 74.7, 43.1, 228. , 202.5, 177. , 293.6, 206.9, 25.1,
       175.1, 89.7, 239.9, 227.2, 66.9, 100.4, 216.4, 182.6, 262.7,
       198.9, 7.3, 136.2, 210.8, 210.7, 53.5, 261.3, 239.3, 102.7,
       131.1, 69. , 31.5, 139.3, 216.8, 199.1, 109.8, 26.8, 129.4,
       213.4, 16.9, 27.5, 120.5, 5.4, 116. , 76.4, 239.8, 75.3,
       68.4, 213.5, 193.2, 76.3, 110.7, 88.3, 134.3, 28.6, 217.7,
       250.9, 107.4, 163.3, 197.6, 184.9, 289.7, 135.2, 222.4, 296.4,
       280.2, 187.9, 238.2, 137.9, 25. , 90.4, 13.1, 255.4, 225.8,
       241.7, 175.7, 209.6, 78.2, 75.1, 139.2, 125.7, 19.4, 141.3,
       18.8, 224. , 123.1, 229.5, 87.2, 7.8, 80.2, 220.3, 59.6,
       0.7, 265.2, 8.4, 219.8, 36.9, 48.3, 25.6, 273.7, 43. ,
       73.4, 193.7, 220.5, 104.6, 96.2, 140.3, 243.2, 38. , 44.7,
       280.7, 121. , 171.3, 187.8, 4.1, 93.9, 149.8, 11.7, 131.7,
       172.5, 85.7, 188.4, 163.5, 117.2, 234.5, 17.9, 206.8, 215.4,
       284.3, 50. , 164.5, 19.6, 168.4, 276.9, 248.4, 170.2, 276.7,
       165.6, 156.6, 218.5, 56.2, 287.6, 253.8, 205. , 139.5, 191.1,
       286. , 18.7, 39.5, 75.5, 166.8, 149.7, 38.2, 94.2, 283.6,
       232.1])
```

In [10]:

```
a.groupby(['radio']).count()
```

Out[10]:

	Unnamed: 0	TV	newspaper	sales
--	------------	----	-----------	-------

radio

0.0	1	1	1	1
0.3	1	1	1	1
0.4	1	1	1	1
0.8	1	1	1	1
1.3	1	1	1	1
...
47.8	1	1	1	1
48.9	2	2	2	2
49.0	2	2	2	2
49.4	2	2	2	2
49.6	1	1	1	1

167 rows × 4 columns

In [13]:

```
b=a.drop(['Unnamed: 0'],axis=1)
b
```

Out[13]:

	TV	radio	newspaper	sales
--	----	-------	-----------	-------

0	230.1	37.8	69.2	22.1
1	44.5	39.3	45.1	10.4
2	17.2	45.9	69.3	9.3
3	151.5	41.3	58.5	18.5
4	180.8	10.8	58.4	12.9
...
195	38.2	3.7	13.8	7.6
196	94.2	4.9	8.1	9.7
197	177.0	9.3	6.4	12.8
198	283.6	42.0	66.2	25.5
199	232.1	8.6	8.7	13.4

200 rows × 4 columns

In [14]:

```
b.shape
```

Out[14]:

```
(200, 4)
```

In [16]:

```
c=pd.get_dummies(b,dtype=int)
c
```

Out[16]:

	TV	radio	newspaper	sales
0	230.1	37.8	69.2	22.1
1	44.5	39.3	45.1	10.4
2	17.2	45.9	69.3	9.3
3	151.5	41.3	58.5	18.5
4	180.8	10.8	58.4	12.9
...
195	38.2	3.7	13.8	7.6
196	94.2	4.9	8.1	9.7
197	177.0	9.3	6.4	12.8
198	283.6	42.0	66.2	25.5
199	232.1	8.6	8.7	13.4

200 rows × 4 columns

In [26]:

```
cor=b.corr()
cor
```

Out[26]:

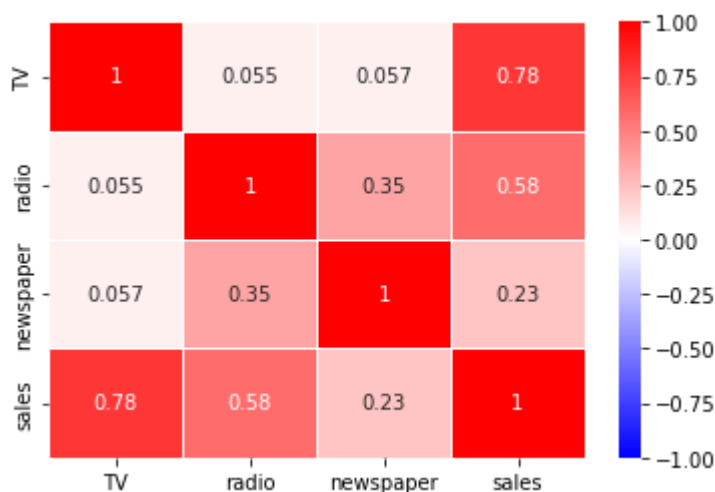
	TV	radio	newspaper	sales
TV	1.000000	0.054809	0.056648	0.782224
radio	0.054809	1.000000	0.354104	0.576223
newspaper	0.056648	0.354104	1.000000	0.228299
sales	0.782224	0.576223	0.228299	1.000000

In [27]:

```
import seaborn as sns
sns.heatmap(cor,vmax=1,vmin=-1,annot=True,linewidths=.5,cmap='bwr')
```

Out[27]:

<AxesSubplot:>



In [28]:

```
y=b['sales']
y
```

```
Out[28]: 0      22.1
          1      10.4
          2       9.3
          3      18.5
          4      12.9
          ...
        195       7.6
        196       9.7
        197      12.8
        198      25.5
        199      13.4
        Name: sales, Length: 200, dtype: float64
```

```
In [29]: x=b.drop(['sales'],axis=1)
          x
```

```
Out[29]:
```

	TV	radio	newspaper
0	230.1	37.8	69.2
1	44.5	39.3	45.1
2	17.2	45.9	69.3
3	151.5	41.3	58.5
4	180.8	10.8	58.4
...
195	38.2	3.7	13.8
196	94.2	4.9	8.1
197	177.0	9.3	6.4
198	283.6	42.0	66.2
199	232.1	8.6	8.7

200 rows × 3 columns

```
In [30]: from sklearn.model_selection import train_test_split
          x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.33,random_state=56)
```

```
In [31]: x_test.head()
```

```
Out[31]:
```

	TV	radio	newspaper
71	109.8	14.3	31.7
42	293.6	27.7	1.8
82	75.3	20.3	32.5
167	206.8	5.2	19.4
186	139.5	2.1	26.6

```
In [34]: y_test.head()
```

```
Out[34]: 71      12.4
         42      20.7
         82      11.3
         167     12.2
         186     10.3
         Name: sales, dtype: float64
```

```
In [36]: from sklearn.linear_model import LinearRegression
         reg=LinearRegression()
         reg.fit(x_train,y_train)
```

```
Out[36]: LinearRegression()
```

```
In [38]: ypred=reg.predict(x_test)
         ypred
```

```
Out[38]: array([10.4756719 , 21.82165244, 10.09454737, 13.20799486,  9.45599683,
                23.01113699,  9.87650264, 15.07017528, 16.81694846, 20.11338747,
                12.60849442, 14.22239707, 14.28413833, 10.31579203, 13.46451076,
                10.52216123, 20.17178192, 10.36372386,  9.65941317, 17.48838889,
                13.98623695,  7.60987808, 12.6920567 , 22.51493404, 14.4554371 ,
                18.87211514, 19.12663059,  9.09933506, 16.27205158, 12.35277988,
                 4.23270305, 11.87633558, 13.68612314, 23.97751066, 15.33776094,
                10.0214975 ,  9.93746159, 12.82159604,  4.3446325 , 14.52873797,
                 9.02046554,  6.47163978, 17.2404426 , 19.80858679, 20.86124068,
                18.24603228, 16.92056626, 22.72849805, 14.75392915, 21.43144583,
                12.23310504, 23.78371511, 17.96973831,  8.76661769,  9.87513484,
                 5.59537846,  6.82183018,  3.39635136, 17.68613738,  5.99632004,
                10.82281985, 17.42604242, 17.14056494,  6.35770938, 22.71573186,
                 9.61443988])
```

```
In [39]: from sklearn.metrics import r2_score
         r2_score(y_test,ypred)
```

```
Out[39]: 0.8716913163812106
```

```
In [42]: from sklearn.metrics import mean_squared_error
         mean_squared_error(ypred,y_test)
```

```
Out[42]: 4.1016448085554265
```

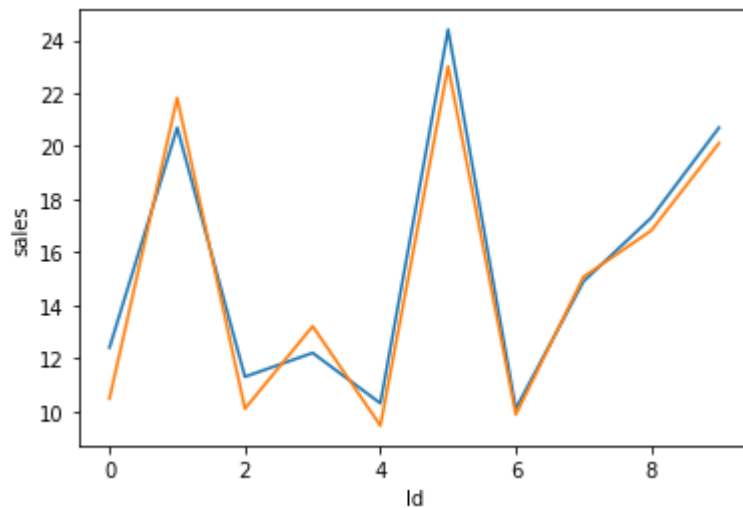
```
In [43]: results=pd.DataFrame(columns=['sales','Predicted'])
         results['sales']=y_test
         results["Predicted"]=ypred
         results=results.reset_index()
         results['Id']=results.index
         results.head(5)
```

```
Out[43]:
```

	index	sales	Predicted	Id
0	71	12.4	10.475672	0
1	42	20.7	21.821652	1
2	82	11.3	10.094547	2
3	167	12.2	13.207995	3
4	186	10.3	9.455997	4

```
In [44]: import seaborn as sns
import matplotlib.pyplot as plt
sns.lineplot(x='Id',y='sales',data=results.head(10))
sns.lineplot(x='Id',y='Predicted',data=results.head(10))
plt.plot()
```

Out[44]: []



In []: